CUDA Implementation of Position Based Fluids

CSC417 Course Project

Qingyuan Qie, Changlin Su

Department of Computer Science
University of Toronto

December 20, 2021

Enforcing incompressibility

For particle i at position p_i, we compute the density of the fluid around particle i using the estimator:
$\rho_{F(i)} = \sum_{j \in F(i)} m_j W_{poly6}(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

Enforcing incompressibility

For particle i at position $\boldsymbol{p}_i$, we compute the density of the fluid around particle i using the estimator:
$\rho_i = \sum_{j \in F(i)} m_j W_{poly6}(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

Then we have the constant density constraint:
$C_i(\boldsymbol{p}) = \frac{\rho_i}{\rho_0} - 1$

Enforcing incompressibility

For particle i at position $\boldsymbol{p}_i$, we compute the density of the fluid around particle i using the estimator:
$\rho_i = \sum_{j \in F(i)} m_j W_{poly6}(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

Then we have the constant density constraint:
$C_i(\boldsymbol{p}) = \frac{\rho_i}{\rho_0} - 1$

And we want to compute a position correction $\Delta\boldsymbol{p}$, such that:
$C(\boldsymbol{p} + \Delta\boldsymbol{p}) = 0$

Position Update from Solving Incompressibility

For particle i at position $\boldsymbol{p}_i$, we have,
$\lambda_i = -\frac{C_i(\boldsymbol{p})}{\sum_k |\nabla_{\boldsymbol{p}_k} C_i|^2}$

Position Update from Solving Incompressibility

For particle i at position $\boldsymbol{p}_i$, we have,
$\lambda_i = -\frac{C_i(\boldsymbol{p})}{\sum_k |\nabla_{\boldsymbol{p}_k} C_i|^2}$

Then the position correction $\Delta\boldsymbol{p}_i$ including affect from neighboring particles is,
$\Delta\boldsymbol{p}_i = \frac{1}{\rho_0} \sum_j (\lambda_i + \lambda_j) \nabla W(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

Position Update from Solving Incompressibility

For particle i at position $\boldsymbol{p}_i$, we have,
$\lambda_i = -\frac{C_i(\boldsymbol{p})}{\sum_k |\nabla_{\boldsymbol{p}_k} C_i|^2}$

Then the position correction $\Delta \boldsymbol{p}_i$ including affect from neighboring particles is,
$\Delta \boldsymbol{p}_i = \frac{1}{\rho_0} \sum_j (\lambda_i + \lambda_j) \nabla W(\boldsymbol{p}_i - \boldsymbol{p}_j, h)$

And the position update is,
$\boldsymbol{p}_i^* = \boldsymbol{p}_i + \Delta \boldsymbol{p}_i$

Simulation Step

Algorithm 1 simulation step

for all particles i **do**: ▷ Fluid advect
    apply forces $\boldsymbol{v}_i = \boldsymbol{v}_i + \Delta t f_{ext}$
    predict position $\boldsymbol{x}_i^* = \boldsymbol{x}_i + \Delta t \boldsymbol{v}_i$
end for
for all particles i **do**:
    find neighboring particles $F(\boldsymbol{x}_i^*)$
end for
while *iter* $<$ *solverIterations* **do**: ▷ Iterativly solves imcompressibility constraints
    for all particles i **do**:
        calculate λ_i
    end for
    for all particles i **do**:
        calculate $\Delta \boldsymbol{p}_i$
        perform collision detection and response
    end for
    for all particles i **do**:
        update position $\boldsymbol{x}_i^* = \boldsymbol{x}_i^* + \Delta \boldsymbol{p}_i$
    end for
end while
for all particles i **do**:
    update velocity $\boldsymbol{v}_i = \frac{1}{\Delta t}(\boldsymbol{x}_i^* - \boldsymbol{x}_i)$
    apply viscosity
    update position $\boldsymbol{x}_i = \boldsymbol{x}_i^*$
end for

Performance of Implementation

References

[1] Nvidia Corp. 2021. *CUDA OpenGL Interoperability*. https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__OPENGL.html

[2] Nvidia Corp. 2021. *Cuda Samples, Particles*. https://github.com/NVIDIA/cuda-samples/tree/master/Samples/particles

[3] Simon Green. 2010. *Particle Simulation using CUDA*. https://developer.download.nvidia.com/assets/cuda/files/particles.pdf

[4] Alec Jacobson. 2021. *libigl*. https://github.com/libigl/libigl/

[5] Miles Macklin and Matthias Müller. 2013. Position Based Fluids. *ACM Trans. Graph.* 32, 4, Article 104 (jul 2013), 12 pages. https://doi.org/10.1145/2461912.2461984

[6] Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-Based Fluid Simulation for Interactive Applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 154–159.

[7] Keren Zhu. 2018. *The Simulation, Rendering and Application of Incompressible Fluids*. https://github.com/naeioi/PBF-CUDA/blob/master/paper/simu-fluids-keren.pdf